Number Path

In this assignment, you will write a program that finds a path through a grid of numbers using a technique called *depth-first search*.

As input, you will be given a grid of numbers, a start point, an end point, and a target sum. Your task is to find a path that moves orthogonally through the grid, keeping a running total of the numbers along the path, and ending at the end point with the required target sum.

Details:

- You can assume that a number grid will be no larger than 10 by 10. An example of a number grid might look like the following:

```
34  58  12  10  34
 3  91  10  10  41
10  76  10   7  12
10  82  10  81  98
10  10  10   9  17
```

- The start point will be specified with two numbers, the row number and the column number. Note that when you count rows and columns, you start with zero. Consequently, in the example grid below, we might specify the start point to be row 2, column 0. This would indicate the number 10 directly under the 3.
- Similarly, the end point will be specified with a row number and a column number. An end point of row 0, column 3 would point to the 10 in the top row between the 12 and 34.
- A target sum is just an integer value that you want your path to sum up to.

If you were given the grid above, with start point (2,0), end point (0,3), and target sum of 100, then you can find a successful path by following the ten 10s in the grid.

Input:

An input file will contain the following:

- **First line:** 7 integers
    1. targetValue, the target sum
    2. grid_rows, the number of rows in the grid
    3. grid_cols, the number of columns in the grid
    4. start_row, the row number of the start point
    5. start_col, the column number of the start point
    6. end_row, the row number of the end point
    7. end_col, the column number of end point

- **All subsequent lines:** there will be (grid_rows) additional lines in the input file. Each line will consist of (grid_cols) integers representing the numbers in the grid for that row.

Here are three examples of input files in which your program should successfully find paths:

- pathdata1: this file has a path that is not obvious to find.
- pathdata2: this has a smaller, rectangular grid.
- pathdata3: this has a larger grid with several dead-end paths. This file is an excellent example of how you can design complex mazes and solve them using your program!

Hints:

- Your main program should do the following tasks:
  - Open the input file "pathdata.txt".
  - Read the contents of the first line into variables.
  - Read in the grid. I recommend representing it as a list of lists.
  - Define a class "State", which has as instance variables a grid, a path history, a start row and column, and a sum.
  - Create an instance of class State, assigning appropriate values to its instance variables.
  - Print out the values of the variables and the grid in a nice format. This will ensure you read everything in correctly and built your data structure the way you wanted it. I **strongly** recommend you define a nice __str__ method for class State to use here, to show progress as your program executes, and to help you debug.
  - Call a function "solve", described below.

- You should also have a function "solve" which takes a State instance as an argument and returns the solution path, if it finds one, or "None" if it doesn't. "solve" should do the following tasks:
  - Test to see if the State instance is a goal state, meaning it's currently at the end state and the sum matches the target sum. If so, print an appropriate message and show the path history.
  - If it's not a goal state, check to see if the sum exceeds the target sum. If so, print a message and return "None".
  - Try moving right, if doing so is a legal move. Create a new State instance with the appropriate start row/column. Set the current grid point to "None" (to make it as "already visited"). Update the sum and history. Print the new State instance, then recursively call "solve" using the instance as an argument. If the recursive call returns a successful path, return the result.
  - If moving right doesn't work, try moving up, then down, then left.
  - If none of the attempts succeed, return "None".

- It would probably be helpful for you to write a function "isValid" that lets you know if a proposed move is a valid one. isValid would take as arguments a current grid, its size, and a proposed row and column position, and returns True if it's a valid position, False if it isn't. The position would be invalid if you're either trying to move outside the grid boundary, or if you were trying to move to a location already visited by your current path (meaning the location has value "None").

Output:

Your output should display the original problem, a snapshot of the grid after every proposed move, and a final result. Sample output corresponding to input file "pathdata1" appears as below:

```
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   17   29   19
      None 5    11   42   14
   history: [8]
   start point: (4,0)
   sum so far: 8

Is this a goal state?
No.  Can I move right?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   17   29   19
      None None 11   42   14
   history: [8, 5]
   start point: (4,1)
   sum so far: 13

Is this a goal state?
No.  Can I move right?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   17   29   19
      None None None 42   14
   history: [8, 5, 11]
   start point: (4,2)
   sum so far: 24
```

```
Is this a goal state?
No.  Can I move right?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   17   29   19
      None None None None 14
   history: [8, 5, 11, 42]
   start point: (4,3)
   sum so far: 66

Is this a goal state?
No.  Can I move right?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   17   29   19
      None None None None None
   history: [8, 5, 11, 42, 14]
   start point: (4,4)
   sum so far: 80

Is this a goal state?
No.  Can I move right?
No.  Can I move up?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   17   29   None
      None None None None None
   history: [8, 5, 11, 42, 14, 19]
   start point: (3,4)
   sum so far: 99

Is this a goal state?
No.  Can I move right?
No.  Can I move up?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   None
      28   12   17   29   None
```

```
      None None None None None
   history: [8, 5, 11, 42, 14, 19, 3]
   start point: (2,4)
   sum so far: 102

Is this a goal state?
No. Target exceeded:  abandoning path
No.  Can I move down?
No.  Can I move left?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   17   None None
      None None None None None
   history: [8, 5, 11, 42, 14, 19, 29]
   start point: (3,3)
   sum so far: 128

Is this a goal state?
No. Target exceeded:  abandoning path
Couldn't move in any direction.  Backtracking.
No.  Can I move down?
No.  Can I move left?
Couldn't move in any direction.  Backtracking.
No.  Can I move up?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   17   None 19
      None None None None 14
   history: [8, 5, 11, 42, 29]
   start point: (3,3)
   sum so far: 95

Is this a goal state?
No.  Can I move right?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   17   None None
      None None None None 14
   history: [8, 5, 11, 42, 29, 19]
   start point: (3,4)
   sum so far: 114
```

```
Is this a goal state?
No. Target exceeded:  abandoning path
No.  Can I move up?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   None 3
      28   12   17   None 19
      None None None None 14
   history: [8, 5, 11, 42, 29, 18]
   start point: (2,3)
   sum so far: 113

Is this a goal state?
No. Target exceeded:  abandoning path
No.  Can I move down?
No.  Can I move left?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   None None 19
      None None None None 14
   history: [8, 5, 11, 42, 29, 17]
   start point: (3,2)
   sum so far: 112

Is this a goal state?
No. Target exceeded:  abandoning path
Couldn't move in any direction.  Backtracking.
No.  Can I move down?
No.  Can I move left?
Couldn't move in any direction.  Backtracking.
No.  Can I move up?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   None 29   19
      None None None 42   14
   history: [8, 5, 11, 17]
   start point: (3,2)
   sum so far: 41

Is this a goal state?
No.  Can I move right?
Yes!
```

```
Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   None None 19
      None None None 42   14
   history: [8, 5, 11, 17, 29]
   start point: (3,3)
   sum so far: 70

Is this a goal state?
No.  Can I move right?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   3
      28   12   None None None
      None None None 42   14
   history: [8, 5, 11, 17, 29, 19]
   start point: (3,4)
   sum so far: 89

Is this a goal state?
No.  Can I move right?
No.  Can I move up?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   6
      10   42   36   18   None
      28   12   None None None
      None None None 42   14
   history: [8, 5, 11, 17, 29, 19, 3]
   start point: (2,4)
   sum so far: 92

Is this a goal state?
No.  Can I move right?
No.  Can I move up?
Yes!

Problem is now:
   Grid:
      15   26   16   22   2
      9    32   23   25   None
      10   42   36   18   None
      28   12   None None None
      None None None 42   14
   history: [8, 5, 11, 17, 29, 19, 3, 6]
   start point: (1,4)
   sum so far: 98
```

```
Is this a goal state?
No.  Can I move right?
No.  Can I move up?
Yes!

Problem is now:
   Grid:
      15   26   16   22   None
      9    32   23   25   None
      10   42   36   18   None
      28   12   None None None
      None None None 42   14
   history: [8, 5, 11, 17, 29, 19, 3, 6, 2]
   start point: (0,4)
   sum so far: 100

Is this a goal state?
Solution found!
[8, 5, 11, 17, 29, 19, 3, 6, 2]
>>>
```